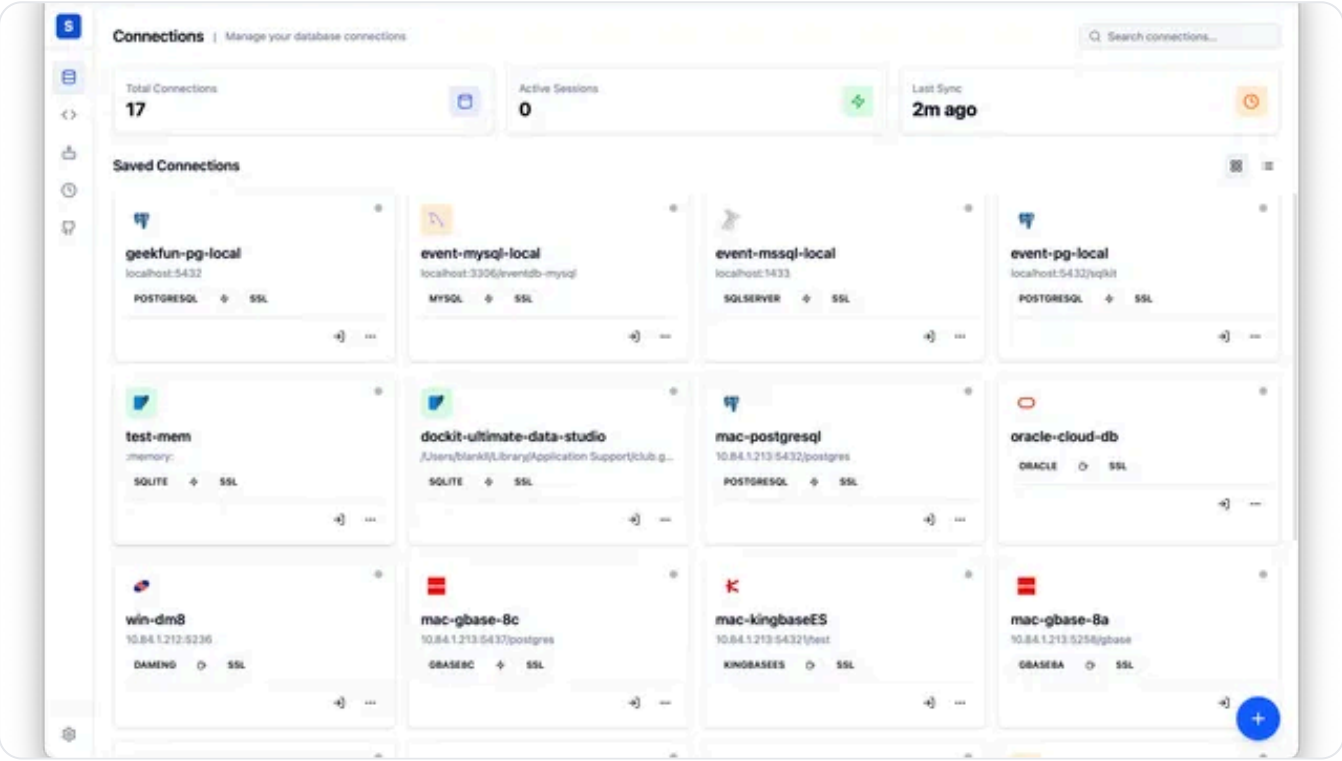
SqlKit

Open-source agentic SQL desktop GUI client for 50+ databases with AI help, privacy-first local credentials, and Rust + Tauri performance.



About

SqlKit is an AI-powered SQL desktop GUI client for 50+ databases that helps developers browse schemas, write queries, and debug SQL from one app. It combines agentic query assistance with a privacy-first design, so your credentials stay on your machine while you work across PostgreSQL, MySQL, SQL Server, Oracle, SQLite, DuckDB, ClickHouse, Snowflake, BigQuery, and many more systems. Built with Rust and Tauri, it aims to feel fast, native, and practical for day-to-day database work.

What is SqlKit?

SqlKit is an open-source agentic SQL client built as a desktop GUI for working with more than 50 databases. It is designed for developers who need one SQL desktop GUI client that can connect to common engines like PostgreSQL, MySQL, SQL Server, Oracle, SQLite, DuckDB, ClickHouse, Snowflake, and BigQuery, plus many additional systems through adapters and bridges.

The app is positioned as an AI-powered database workspace rather than a simple query window. It reads schema context, helps draft SQL, explains execution plans, and can assist with error fixing. SqlKit also emphasizes privacy-first operation: you bring your own LLM key, and credentials stay local on your machine.

Key features

SqlKit combines a traditional SQL desktop GUI client with an agentic assistant and a broad connectivity layer.

- **50+ database support:** Native Rust drivers cover PostgreSQL, MySQL, SQL Server, SQLite, DuckDB, ClickHouse, Oracle, and Firebird.
- **Adapter-based connectivity:** It also supports PG-wire and MySQL-wire compatible engines, JDBC-linked databases, and HTTP bridges for tools like Trino and Presto.
- **AI-powered assistant:** The agent can read your schema, write queries, optimize slow SQL, explain execution plans, and help correct errors.
- **Monaco-based editor:** The SQL editor includes syntax highlighting, context-aware autocomplete, multi-tab editing, and configurable formatting rules by dialect.
- **Schema browser:** A tree view surfaces databases, schemas, tables, columns, indexes, views, procedures, and functions.
- **Data inspection:** You get a virtual-scrolled grid with sort, filter, inline editing, row-level search, and export to CSV, JSON, or Markdown.
- **Visualization tools:** SqlKit renders ER diagrams and query execution plans so relationships and cost hotspots are easier to inspect.
- **Security controls:** SSH tunnels, SSL/TLS, OS keychain storage, and auto-reconnect are built in.

How to use it

SqlKit

Using SqlKit is meant to be straightforward for anyone already comfortable with SQL desktop GUI client workflows.

Download and install the desktop app. The product is built with Rust + Tauri v2, so it is a native desktop app rather than an Electron app.

Create a connection. Choose the database type or adapter path that matches your server, then add the host, credentials, and security settings.

Open the schema browser. Use the tree view to inspect databases, schemas, tables, columns, and relationships before writing SQL.

Write or refine queries. The Monaco editor supports multi-tab work, autocomplete, and formatting, which helps when switching between several SQL desktop GUI client tasks.

Use the agentic assistant. Ask it to draft a query, explain a plan, or fix a failing statement using the schema context already available in the app.

Inspect results and structure. Review data in the grid, export it if needed, and switch to ER diagrams or execution-plan views when you need more context.

Who it is for

SqlKit is best suited to developers, data engineers, and analysts who want one SQL desktop GUI client for many database engines. It is especially useful if you work across mixed environments and do not want to maintain separate tools for each system.

This app fits teams that care about local control and security, because it keeps credentials on the machine and supports SSH, SSL/TLS, and OS keychain storage. It is also a good match for people who want AI help inside a SQL desktop GUI client without sending everything to a hosted service by default.

Typical use cases include:

- exploring unfamiliar schemas before writing queries
- debugging slow SQL with execution-plan views
- browsing data and editing rows directly in a grid
- connecting to several database families from one interface
- using AI to speed up query drafting and troubleshooting

What to know before you use it

SqlKit is broad in database coverage, but it is still a desktop SQL desktop GUI client and not a hosted database platform. You still need valid access to the databases you connect to, and adapter support may vary by engine and connection method.

A few caveats are important:

- AI features depend on your own LLM provider key and the provider you choose, such as OpenAI, Anthropic, DeepSeek, or Ollama.
- The assistant can help read schema and suggest SQL, but it cannot guarantee correctness, safety, or optimal performance for every query.
- Destructive actions should still be reviewed carefully, even with safety gates enabled.
- JDBC and bridge-based connections can require extra driver or JRE handling depending on the target database.
- For production changes, treat the SQL desktop GUI client as a tool for inspection and assistance, not as a substitute for your database's official admin or audit processes.

If you need a privacy-first SQL desktop GUI client with broad database support and built-in AI assistance, SqlKit is a strong option. It combines the essentials of a SQL desktop GUI client with local credential handling, schema-aware help, and multi-database coverage in one app.

FAQ

What databases does SqlKit support?

Is SqlKit free to use?

How does the AI agent work?

Which databases does SqlKit support?

Does SqlKit keep credentials local?

Can I transfer data between different databases?

What platforms does SqlKit run on?

What operating systems can run SqlKit?

How does SqlKit keep my data secure?

Who should use SqlKit instead of a single-database tool?

Sponsored

Belay – Awa... Featured

Developer Tools

Stays awake while your agents work.

Free # Developer Tools

View ↗

Partly Featured

Finance

Partly is a mobile app that lets friends quickly scan receipts, split any bill fairl...

Free # Finance

View ↗

Visit website ↗

sqlkit

Pricing Free

Category AI

Launched	Aug 2026
Updated	18 days ago
Verified	1 months ago

TAGS

ai # sql # database-client # postgresql # database # desktop-gui # desktop-app # ai assistant # data-visualization # rust # tauri # ai agent # cross-platform # open source # query-editor # cross-engine

© 2026 Dir Hub

SqlKit